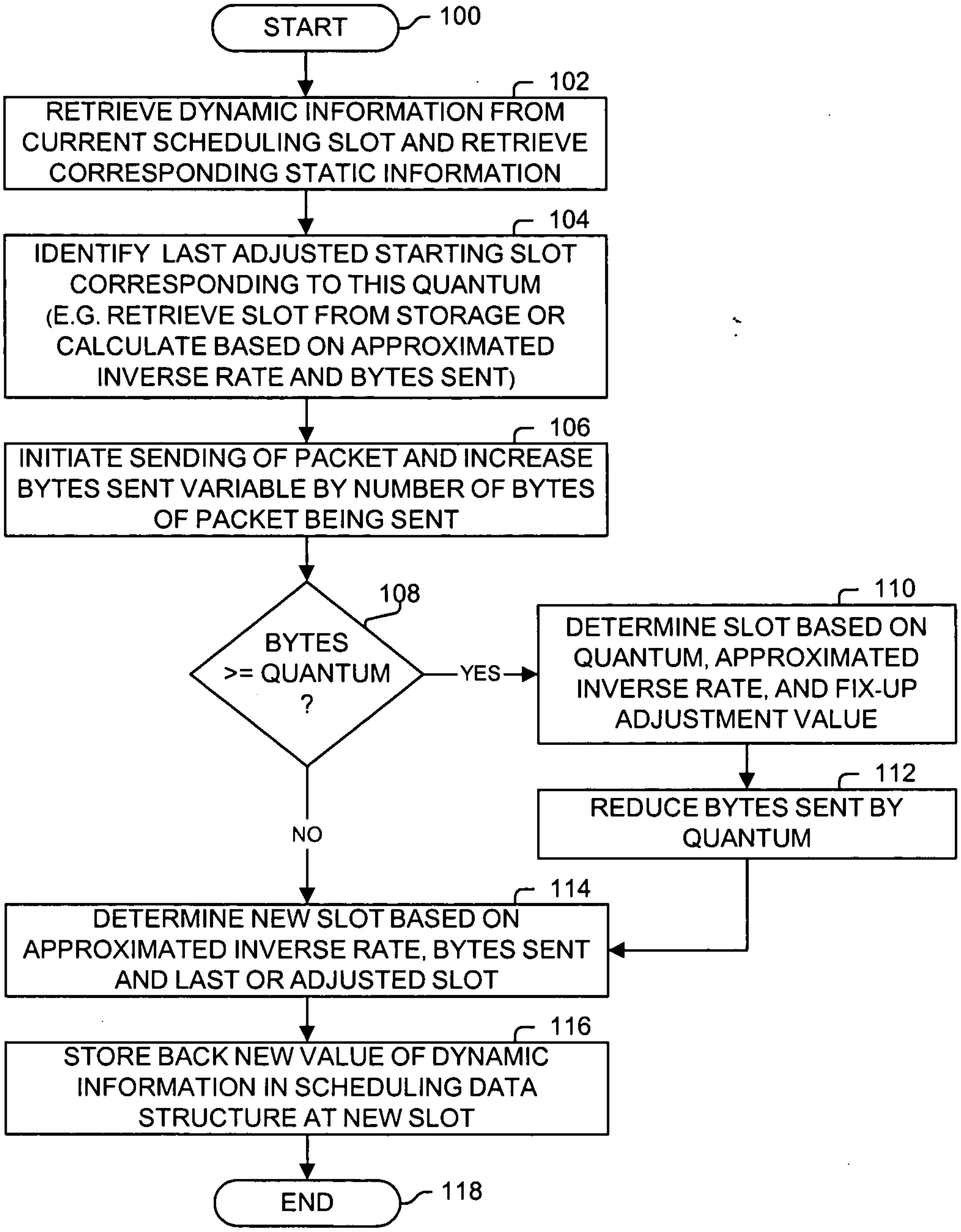
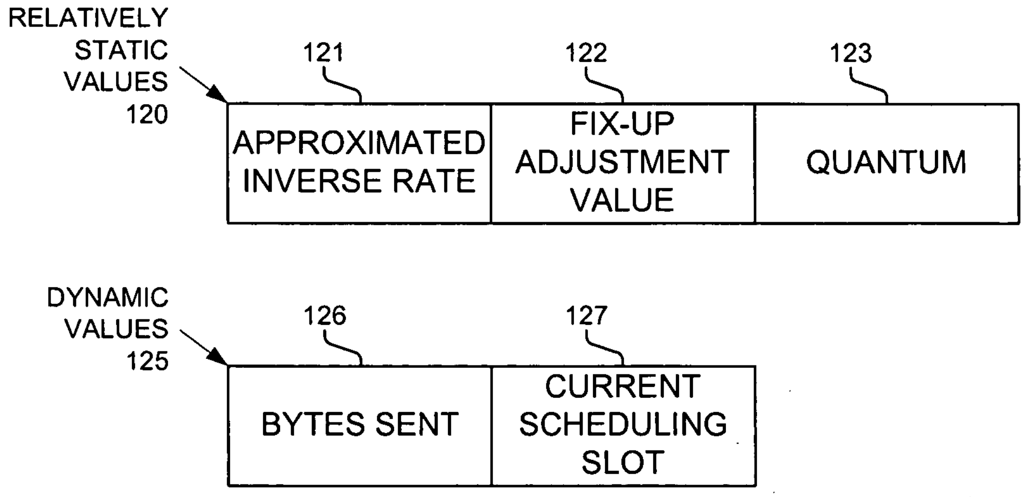

START
100
102
RETRIEVE DYNAMIC INFORMATION FROM CURRENT SCHEDULING SLOT AND RETRIEVE CORRESPONDING STATIC INFORMATION
104
IDENTIFY LAST ADJUSTED STARTING SLOT CORRESPONDING TO THIS QUANTUM (E.G. RETRIEVE SLOT FROM STORAGE OR CALCULATE BASED ON APPROXIMATED INVERSE RATE AND BYTES SENT)
106
INITIATE SENDING OF PACKET AND INCREASE BYTES SENT VARIABLE BY NUMBER OF BYTES OF PACKET BEING SENT
108
BYTES >= QUANTUM ?
YES
110
DETERMINE SLOT BASED ON QUANTUM, APPROXIMATED INVERSE RATE, AND FIX-UP ADJUSTMENT VALUE
112
REDUCE BYTES SENT BY QUANTUM
NO
114
DETERMINE NEW SLOT BASED ON APPROXIMATED INVERSE RATE, BYTES SENT AND LAST OR ADJUSTED SLOT
116
STORE BACK NEW VALUE OF DYNAMIC INFORMATION IN SCHEDULING DATA STRUCTURE AT NEW SLOT
END
118

FIGURE 1A



STORED VALUES
FIGURE 1B

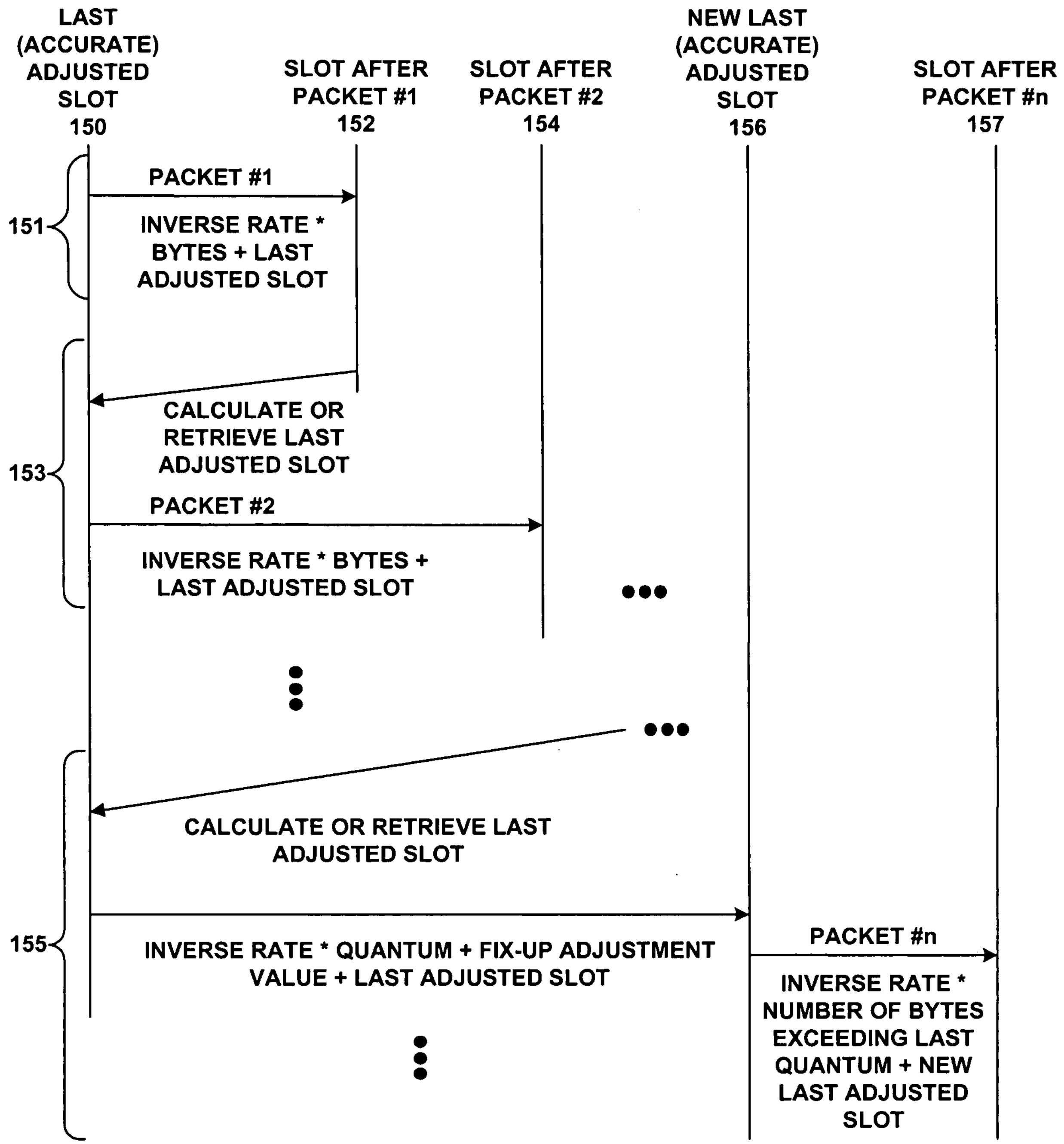
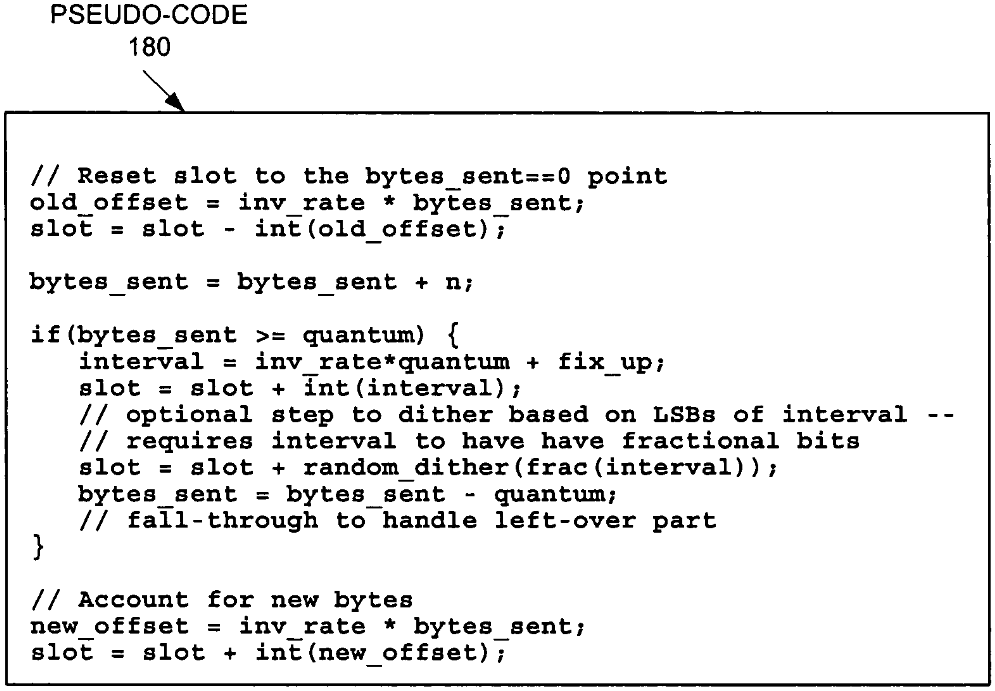

LAST (ACCURATE) ADJUSTED SLOT 150
SLOT AFTER PACKET #1 152
SLOT AFTER PACKET #2 154
NEW LAST (ACCURATE) ADJUSTED SLOT 156
SLOT AFTER PACKET #n 157
151
PACKET #1
INVERSE RATE * BYTES + LAST ADJUSTED SLOT
153
CALCULATE OR RETRIEVE LAST ADJUSTED SLOT
PACKET #2
INVERSE RATE * BYTES + LAST ADJUSTED SLOT
155
CALCULATE OR RETRIEVE LAST ADJUSTED SLOT
INVERSE RATE * QUANTUM + FIX-UP ADJUSTMENT VALUE + LAST ADJUSTED SLOT
PACKET #n
INVERSE RATE * NUMBER OF BYTES EXCEEDING LAST QUANTUM + NEW LAST ADJUSTED SLOT

FIGURE 1C


PSEUDO-CODE
180

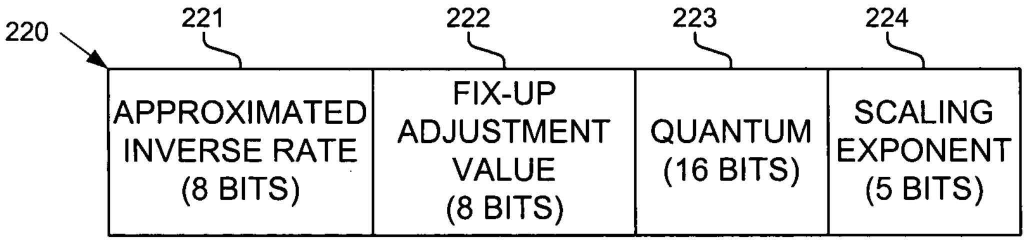
```
// Reset slot to the bytes_sent==0 point
old_offset = inv_rate * bytes_sent;
slot = slot - int(old_offset);

bytes_sent = bytes_sent + n;

if(bytes_sent >= quantum) {
   interval = inv_rate*quantum + fix_up;
   slot = slot + int(interval);
   // optional step to dither based on LSBs of interval --
   // requires interval to have have fractional bits
   slot = slot + random_dither(frac(interval));
   bytes_sent = bytes_sent - quantum;
   // fall-through to handle left-over part
}

// Account for new bytes
new_offset = inv_rate * bytes_sent;
slot = slot + int(new_offset);
```

FIGURE 1D


220
221
222
223
224
APPROXIMATED INVERSE RATE (8 BITS)
FIX-UP ADJUSTMENT VALUE (8 BITS)
QUANTUM (16 BITS)
SCALING EXPONENT (5 BITS)

RELATIVELY STATIC STORED VALUES FOR RATE COMPUTATION
FIGURE 2

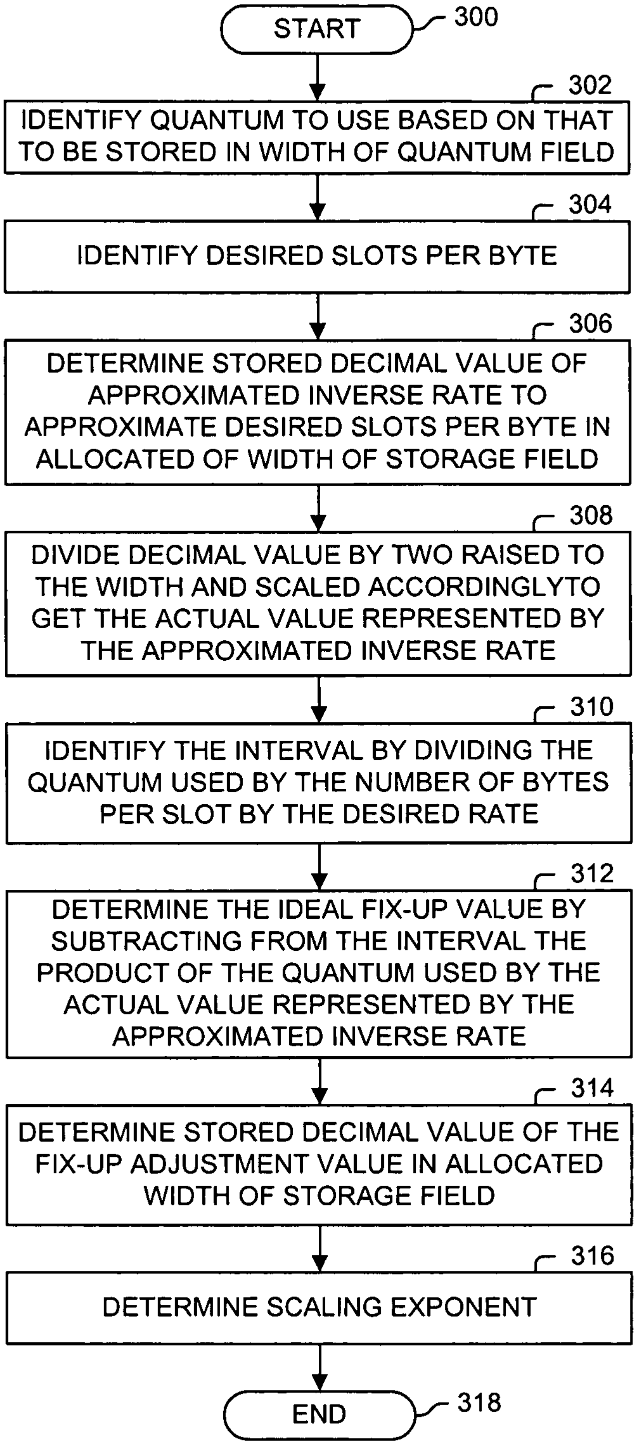

START
300
IDENTIFY QUANTUM TO USE BASED ON THAT TO BE STORED IN WIDTH OF QUANTUM FIELD
302
IDENTIFY DESIRED SLOTS PER BYTE
304
DETERMINE STORED DECIMAL VALUE OF APPROXIMATED INVERSE RATE TO APPROXIMATE DESIRED SLOTS PER BYTE IN ALLOCATED OF WIDTH OF STORAGE FIELD
306
DIVIDE DECIMAL VALUE BY TWO RAISED TO THE WIDTH AND SCALED ACCORDINGLYTO GET THE ACTUAL VALUE REPRESENTED BY THE APPROXIMATED INVERSE RATE
308
IDENTIFY THE INTERVAL BY DIVIDING THE QUANTUM USED BY THE NUMBER OF BYTES PER SLOT BY THE DESIRED RATE
310
DETERMINE THE IDEAL FIX-UP VALUE BY SUBTRACTING FROM THE INTERVAL THE PRODUCT OF THE QUANTUM USED BY THE ACTUAL VALUE REPRESENTED BY THE APPROXIMATED INVERSE RATE
312
DETERMINE STORED DECIMAL VALUE OF THE FIX-UP ADJUSTMENT VALUE IN ALLOCATED WIDTH OF STORAGE FIELD
314
DETERMINE SCALING EXPONENT
316
END
318

FIGURE 3A

TARGET RATE (%)	ACTUAL RATE (%)	INTERVAL	QUANTUM BYTES	SLOTS / BYTE	APPROX. INVERSE RATE	FIX-UP ADJUSTMENT VALUE
640	640.00689	1.020000	65280	0.0000156	131/256 * 2^{-15}	18/256 * 2^{-7}
300	300.00330	1.092267	32768	0.0000333	139/256 * 2^{-14}	207/256 * 2^{-7}
200	200.00034	1.638400	32768	0.0000500	209/256 * 2^{-14}	183/256 * 2^{-7}
100	100.00017	1.638400	16384	0.0001000	209/256 * 2^{-13}	183/256 * 2^{-7}
90	90.00005	1.820444	16384	0.0001111	233/256 * 2^{-13}	4/256 * 2^{-7}
75	75.00083	2.184533	16384	0.0001333	139/256 * 2^{-12}	207/256 * 2^{-6}
50	50.00009	3.276800	16384	0.0002000	209/256 * 2^{-12}	183/256 * 2^{-6}
25	25.00004	6.553600	16384	0.0004000	209/256 * 2^{-11}	183/256 * 2^{-5}
10	10.00013	16.384000	16384	0.0010000	131/256 * 2^{-9}	18/256 * 2^{-3}
1	1.00000	163.840000	16384	0.0100000	163/256 * 2^{-6}	215/256 * 2^{0}
0.1	0.10000	1638.400000	16384	0.1000000	204/256 * 2^{-3}	205/256 * 2^{3}
0.012207	0.0122069	8192.000000	10000	0.8192000	209/256 * 2^{0}	112/256 * 2^{6}

**EXAMPLES OF RATE ENCODING
(10KB MTU, 8192 SLOTS)
FIGURE 3B**

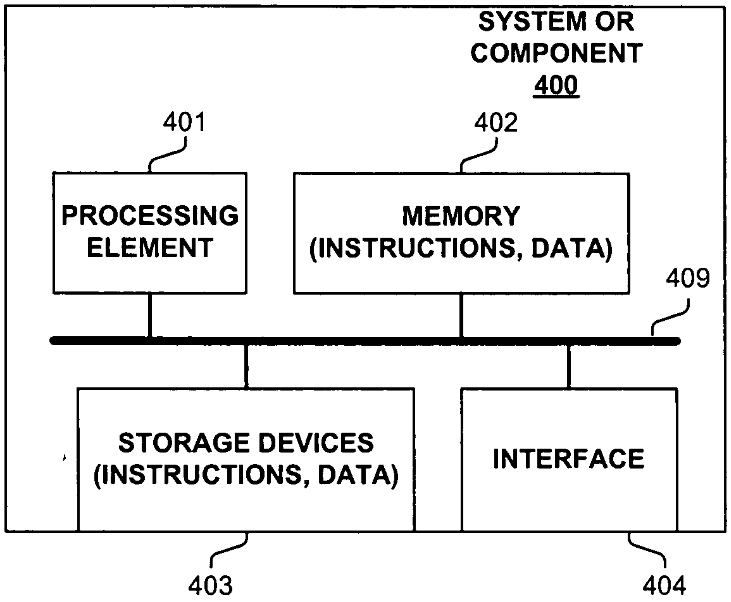

SYSTEM OR COMPONENT 400
401
402
PROCESSING ELEMENT
MEMORY (INSTRUCTIONS, DATA)
409
STORAGE DEVICES (INSTRUCTIONS, DATA)
INTERFACE
403
404

FIGURE 4A

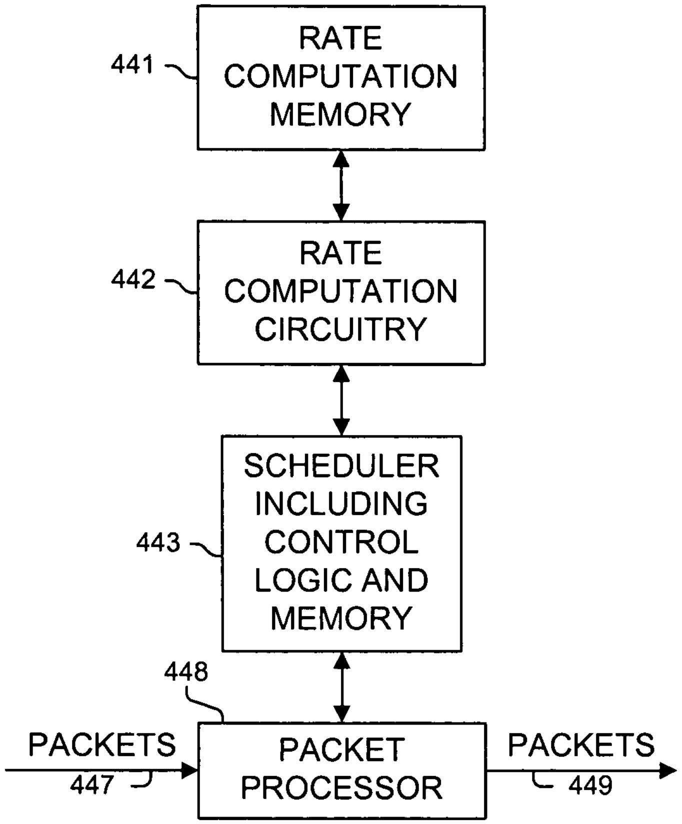

441
RATE COMPUTATION MEMORY
442
RATE COMPUTATION CIRCUITRY
443
SCHEDULER INCLUDING CONTROL LOGIC AND MEMORY
448
PACKETS
447
PACKET PROCESSOR
PACKETS
449

FIGURE 4B